Binary Numbers and Addition Homework

You should refer to the **homework policy** for details on how this homework should be submitted.

Attempt all questions

Question 1

Write down the **largest** binary number that can be held in **8-bits**. Work out what the denary equivalent is.

Binary form
The largest binary number that can be held in 8-bits is **11111111**

Denary Form

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

128+64+32+16+8+4+2+1 = 255

(**2 marks**)

Question 2

How many bits make:

- one **byte** *is made up of **8** bits*
- one **kilobyte** *is made up of **8000** bits*
- one **megabyte** *is made up of **8000000** bits*

(**3 marks**)

Question 3

What are the possible values that **one bit** can take?

One bit can be: (*Colum containing 1 shows value at top*)

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0

(**1 mark**)

Question 4

Convert the **denary** numbers **37** and **84** into binary. Be sure to **show your working**.

37 in binary is...

32	16	8	4	2	1
1	0	0	1	0	1

84 in binary is...

64	**32**	**16**	**8**	**4**	**2**	**1**
1	0	1	0	1	0	0

(**4 marks**)

Question 5

Add the **binary** numbers generated in the previous question together. Be sure to **show your working**.

128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
		1	0	0	1	0	1
	1	0	1	0	1	0	0
0	1	1	1	1	0	0	1

Final answer of - **01111001**

(**2 marks**)

Question 6

Explain what is meant by **overflow error**. Provide an example to help with your explanation.

An overflow error will be presented when the result of a arithmetic operation is to large to be presented.

(**3 marks**)

Question 7

Convert the decimal numbers 8 and 13 into binary. Multiply the binary numbers for 8 and 13, **showing your working**. Then convert the result back to denary to check your answer.

Numbers to binary

8 into binary is…

8	4	2	1
1	0	0	0

13 into binary is…

8	4	2	1
1	1	0	1

Multiplication
Is done by multiplying the bottom number by the top number, and for every number you move across you add on a 0 to the start of the process, repeat this until you have multiplied all the numbers from the bottom binary number into the top binary number.

1000 x 1101

After step 1: **1000**
After step 2: **00000**
After step 3: **100000**
After step 4: **1000000**

64	32	16	8	4	2	1
0	0	0	1	0	0	0
0	0	0	0	0	0	0
0	1	0	0	0	0	0
1	0	0	0	0	0	0

When you add them together going down the colums the final answer is: **1101000**

When you use the table to convert to denary the answer is: **104**

(**4 marks**)

Total 18 marks